THE LSA DATABASE TO DRIVE THE ACCELERATOR SETTINGS

C. Roderick, R. Billen, CERN, Geneva, Switzerland

Abstract

The LHC Software Architecture (LSA), used to operate the particle accelerators at CERN, is dependent on an on-line database to manage both high and low level parameter settings, including their evolution over time. Accelerator optics models, control sequences, reference values, are amongst the other entities being managed within the database. The LSA database can be considered as being located between the operators and the accelerators; therefore performance, availability, and security of the service as well as data integrity are paramount. To meet these requirements the LSA database model has been carefully developed, all database access is tightly controlled and instrumented, business logic is implemented within the database, and there is a semi-automatic integration with other key accelerator databases. Currently 8.6 million settings for some 40 thousand devices of the LEIR, SPS, and LHC accelerators are being effectively managed.

INTRODUCTION

In 2001, the development of homogenous application software for accelerator control at CERN was started, baptized LSA (LHC Software Architecture) [1]. Within this modular, layered and distributed LSA architecture, the data store ensures the data-driven foundation.

The Oracle® database management system (DBMS) was already used for CERN's LEP (Large Electron-Positron Collider) at the end of the 1980-ies [2]. It was only in the last years of LEP operation that the databases were used in an on-line mode. The ultimate efficiency, with which LEP could be operated, was in large part due to the integration of a well designed control system using commercial databases [3].

A similar approach was decided upon for the LHC era. The LSA database is used on-line, effectively between the operators and the accelerators.

OVERVIEW & SCOPE

The LSA software and its database targeted high-level controls for the LHC, but used existing accelerator installations as the test beds. The data model already envisaged a common usage for all CERN accelerators from an early stage. There are many commonalities between accelerators, but also many specifics to individual accelerators which needed to be accounted for. Currently, this controls data is operationally used for:

- LHC (Large Hadron Collider)
- SPS (Super Proton Synchrotron)
- LEIR (Low Energy Ion Ring)
- Transfer lines on SPS and LHC

A simplified view of the data domain (Fig. 1) shows that all of the most important aspects of accelerator controls are covered: *Optics* models and a *Parameter* space with *Settings* in a given *Context* (i.e. one or more *beam processes*, possibly grouped in *magnetic cycles*), to transfer particles across accelerator zones. The domain is generally referred to as *Settings* management, as it can be summarized as run-time parameters that are set to the accelerator equipment (*Devices*).

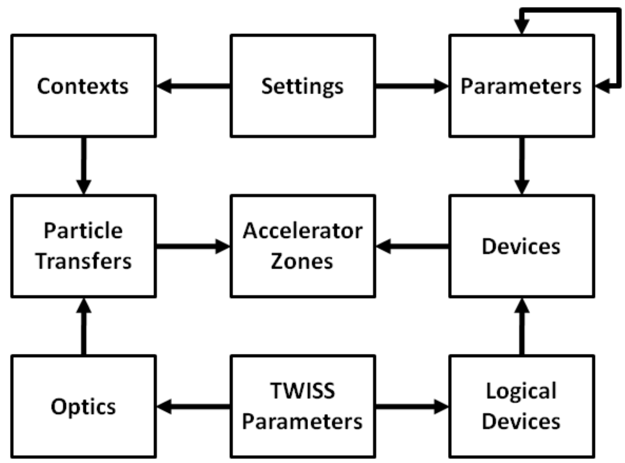

Figure 1: LSA data domain.

In addition, storing the time-stamped history of all settings, as they are tweaked over time by the operators, allows a heuristic approach to accelerator operation.

The current version of the relational model of the database schema comprises 269 tables, with at least as many referential constraints, requiring an A0-size format to visualize the diagram in a readable manner.

DEVELOPMENT ENVIRONMENT

In order to keep tight control on the evolution and quality of the operational deployments, staging across distinct environments is employed. This has resulted in four LSA database schemas for different purposes:

- DEV – Development of new data structures and core software code.
- TEST – Newly developed database objects made available for core software unit testing.
- NEXT – Tested data structures and core software code made available for final integration tests of high level applications using a copy of production data.
- PRO – Production environment for operation with accepted database schema and application software.

New database constructs get deployed in the above order, after agreement with the parties involved. Current PRO data can be copied to the NEXT environment on demand. Since 2005, some 60 major versions of the database schema have been released, corresponding to the growing scope and functionality. All database source code for structures and business logic is managed via the Subversion (SVN) version control facility.

PERFORMANCE

Response times of data queries or manipulations represent the response times of the control system for operators. Performance has been carefully considered from the outset, and designed into the data model. Appropriate object types, column orders, index choices, and use of compression have been selected according to how the data will be used.

The tables containing settings and Twiss parameters are rather large, and continue to grow. Despite this, access times to read and write data must remain acceptable. Table 1 shows the current number of records in a few key tables.

Table 1: No. of Records in Key Tables

Table	No. Records
Contexts	2’316
Setting headers	8’608’880
Setting values	449’905’084
Parameters	67’250
Devices	41’951
Particle Transfers	23
Twiss parameters	1’849’513

Since Oracle DBMS 10*g*, it is extremely important to have up-to-date, representative statistics of the stored data in order to achieve high query performance. This has presented some difficulties, particularly related to having some heavily *skewed* data sets (i.e. asymmetrically distributed). For example, settings data can be skewed across several dimensions: according to the device, parameter, and context required. Solutions have been put in place to ensure acceptable and maintainable query performance.

For key data intensive operations (e.g. cloning of settings for a given context), the business logic has been implemented inside the database using PL/SQL to maximise performance.

Data access from the LSA software core (Java) has been developed in-house, and is highly optimized using appropriate techniques for efficient use of database resources (bind variables, connection pooling, and statement caching), and large transfers of data (bulk inserts and array fetching).

AVAILABILITY

Since the database is used on-line, the availability of the service is paramount. To achieve high-availability, the database is setup as an Oracle Real Application Cluster (RAC). Oracle RAC is a shared database environment where multiple server nodes share DBMS instances, with shared concurrent access to disk. Not only the LSA database, but all accelerator on-line databases [4] are deployed on RAC. The supporting hardware is a cluster of currently 2 servers with full redundancy in all its components. We refer to it as “double everything” (processors, memory, network cards, power supplies, system disks, disk controllers, disk shelves). On several occasions, some components failed, but the service continued to run without any impact on the applications.

Interventions for security patches are performed as *rolling* upgrades via the RAC technology. Database connections are automatically transferred to another server in the cluster, rendering the interference transparent to the operators. Those interventions that do require a database stop, such as a version upgrade or patching of the clusterware, are synchronized with scheduled accelerator maintenance.

SECURITY

The LSA data is highly sensitive, as it acts on high-energy particle beams (e.g. machine critical settings). Maximum data security has to be foreseen in order to avoid erroneous manipulations of the data or the database objects. One single database account holds the database schema and code, for which only the database experts have the credentials to apply modifications. Separate database accounts have different privileges for fine-grained read and/or write data access according to the domain.

Also at the controls software application level, an additional level of security is implemented, called Role-Based Access Control (RBAC). The main motivation of RBAC is to prevent unauthorized access to the accelerator control applications by means of an authentication-authorization scheme [5]. RBAC works by giving people roles and assigning the roles permissions to act on accelerator settings.

In addition, all data changes within key data structures are registered, whereby the previous values together with the details of the modifier are captured.

DATA INTEGRITY

Maintaining data integrity – the assurance that data is consistent and correct – is also vital, in order to prevent controls applications from using corrupted data. The database schema has been designed and implemented according to the relational model. All rules and constraints to enforce the data integrity are implemented within the database, as summarized in Table 2.

Table 2: Data Integrity Constraints in LSA Database

Constraint Type	Count
Unique	343
Referential	299
Value Check	750
Total:	*1392*

SYSTEM INTEGRATION

The LSA database is not a standalone repository, but is part of a distributed information system covering the accelerators.

Data Domains

The LSA *Settings* management is part of the domain of *Operational Data*. In this domain, the run-time data is of the highest importance, covering specific functionalities:

- Accelerator and beam settings (LSA).
- Measurements and long-term Logging.
- Alarms needing operator action.

In our view of the overall accelerator data management, the following adjacent areas are present, in addition to the Operational Data domain:

- Configuration of the Controls System.
- Accelerator and Controls Layout.
- Physical Equipment with maintenance follow-up.

The tight integration of all these data domains is essential for coherent overall data management [4]. Strict quality assurance by means of naming conventions, documentation and procedures is the crucial federating factor.

Inter-Domain Data Propagation

The business logic that enables data propagation between LSA and various other systems within the different data domains has been implemented in Oracle PL/SQL. For example, this allows a semi-automatic integration of data describing accelerator components (from *Layout*) and control devices (from *Controls Configuration*) – including their properties and inter-relationships – into the LSA database. This not only ensures data integrity, but also reduces data entry time.

Similarly, data is automatically propagated from LSA to the database supporting the *Measurement and Logging Service*, to establish required metadata and logging configurations.

Finally, there are some key pieces of information for which there is wide interest in changes of current values (e.g. accelerator mode and status, particle bunch pattern). To avoid the inefficiency of regular polling by all clients, this information is *published* from within the database to the outside world. The implementation uses Oracle Advanced Queuing, which provides a database-integrated message queuing functionality. Via a *subscription* mechanism, the information is further distributed to all interested clients via the normal controls middleware infrastructure [6].

OUTLOOK

The LSA database will continue to grow significantly along several dimensions:

- Increasing history of settings (i.e. run-time trims).
- Increasing number of "knobs" to create handles on specific and finer-grained parameters.
- New database objects (tables) to cater for increased functionality.
- Anticipated scope extension towards other accelerators and installations.

The last point is addressed by the project to renovate the high-level control of the complete injector chain of the LHC [7]. The strategic decision to have a single, unified data model for all controls was made for reasons of maintainability. Consequently, the LSA database will evolve by means of modifications and extensions to the data model. This will also contribute to a considerable augmentation in the data volumes. Performance must remain acceptable for control of all accelerators. Given the mission-criticality, the issues of performance, scalability and availability, are continuously re-evaluated. The introduction of a stand-by database, disk-to-disk backups and innovative commercial solutions are being investigated.

CONCLUSION

The LSA database is an example of a relational database used to effectively manage a vast and complex dataset for high-level accelerator control. The on-line usage does not hinder the real-time requirements of accelerator operations. The exact match between the actual accelerator settings and the image in the database model is guaranteed. The powers of the DBMS have been acknowledged and utilised to the full potential to form a core component of the LHC-era control system.

REFERENCES

[1] G. Kruk *et al.*, "LHC Software Architecture [LSA] – Evolution toward LHC Beam Commissioning", ICALEPCS'07, Knoxville, USA, October 2007, WOPA03, p. 307 (2007)*.

[2] J. Poole, "The Data systems for LEP Control", EPAC 1990, Nice, France, June 1990, p. 857 (1990)*.

[3] M. Lamont, R. Assmann and B. Goddard, "Lessons Learned from LEP", PAC 2001, Chicago, USA, June 2001, RPPH115, p. 3502 (2001)*.

[4] R. Billen *et al.*, "Accelerator Data Foundation: How It All Fits Together", ICALEPCS'09, Kobe, Japan, October 2009, TUB001.

[5] P. Charrue *et al.*, "Role-Based Access Control for the Accelerator Control System at CERN", ICALEPCS'07, Knoxville, USA, October 2007, TPPA04, p. 90 (2007)*.

[6] K. Kostro, R. Billen and C. Roderick, "On-change Publishing of Database Resident Control System Data", ICALEPCS'09, Kobe, Japan, October 2009, TUP013.

[7] S. Deghaye *et al.*, "CERN Proton Synchrotron Complex High-Level Controls Renovation", ICALEPCS'09, Kobe, Japan, October 2009, THA004.

* Published at the Joint Accelerator Conferences Website (JACoW) http://www.JACoW.org